FINAL REPORT:
HLA Secure Combined Federation Architecture
(Part One)

Trusted Information Systems, Inc.
Jarrellann Filsinger

31 December 1996

HLA Secure Combined Federation Architecture

1.0 INTRODUCTION

The Defense Modeling and Simulation Master Plan [3] calls for the establishment of a DoD-wide High Level Architecture (HLA) for modeling and simulation (M&S), applicable to a wide range of functional simulation applications. The Defense Modeling and Simulation Office (DMSO) is responsible for development of standards and an infrastructure to support widespread sharing and reuse within the simulation community. HLA Version 1.0 was approved by the Architecture Management Group (AMG) on 22 August 1996 and by Dr. Paul Kaminski on 10 September 1996 making the HLA the DoD standard for modeling and simulation. General background on the components of the HLA can be found on: www.dmso.mil.

One of the open issues within the HLA that must be addressed is security. Simulations are combined into a logical grouping called a federation; each partner in a federation is called a federate. A federation, as implemented by the prototype federations (protofederations) were implemented at a system high[1] mode of operation. However, due to increased operational requirements to share classified and unclassified data, federations need to exchange data between security domains, or across security boundaries. Security must be addressed when one federate needs to share information with another federate that does not operate at the same security classification level as the first. Classification level is the major characteristic of security domain that might differ, although need-to-know and releasability restrictions can also affect information sharing. For simplicity of exposition, much of the presentation of issues in this paper will be phrased in terms of hierarchical sensitivity level, although requirements exist with need-to-know and releasability. The solution proposed can be configured to meet any of these requirements.

One of the challenges is to provide system security mechanisms and features within the HLA framework without affecting the basic HLA properties and services. This paper contains a description of a two-way information flow guard that will allow appropriately limited communication between federates having different security requirements and the process by which federations can be designed to use that guard.

1.1 Report Outline

This report contains 5 Sections. Section 1 is the introduction. Section 2 provides a description of the security concept of operations for HLA Combined Federations, then Section 3 discusses the HLA Secure Combined Federation architecture. Section 4 focuses on the critical security aspects of the architecture -- the HLA security guard. Section 5 contains references used in the report.

---

1 System High mode of operation requires all personnel to be cleared for all information in the federation, formal access approval, and a valid need-to-know for some of the information within the federation.

2.0 SECURITY CONCEPT OF OPERATIONS

A series of HLA security architectures has been developed to meet near-term, mid-term, and long-term simulation functional requirements.[5] The near-term security architecture addresses the requirement for federations to process information in a system high mode of operation and the long-term security architecture discusses the federation in a multilevel secure mode of operation. The HLA Secure Combined Federation architecture is the mid-term architecture that is expanded in this report. The security concept of operations presented in the following sections includes a high level view of the information flow requirements, the construction of a multidomain federation, or Combined Federation, and assumptions and constraints. Assurance attributes of the security guard, which is trusted to transfer information from one security domain to another, are also discussed.

In this phase of the HLA security architecture development, the concept of operations is in progress. Information is still needed as to the security environments for the guard, specific operational details for guard administration, the security clearance levels of the guard operators and other items. Other developments at the architectural level such as the need for more than one guard, communication security requirements, and identification and authentication requirements are being explored. The following sections detail the generally recognized concepts, assumptions, and constraints at this point in the architecture development.

2.1 Federation Construction

A simulation exercise that has functional requirements for data at different security levels will comprise a federation consisting of multiple security domains, where each security domain will run at a system high security mode of operation. Federations are constructed using the Federation Development and Execution Process Model developed by DMSO. [16] (For additional information on the security engineering in the Federation Development and Execution Process Model, see Part II of this report.) Sharing data among federations requires the development of a Federation Object Model (FOM). The purpose of a FOM "is to provide a specification of the exchange of all public data among federates in a common, standardized format. The content of this public data includes 1) an enumeration of all public object classes, 2) a description of all interaction types and associated parameters, and 3) a specification of the attributes that characterize the public objects." [14]

The joining of several system high federations is a Combined Federation. A Combined Federation will be designed from several single security level FOMs. These FOMs may be developed for the Combined Federation, or they may already exist in the Modeling and Simulation Resource Repository (MSRR). A Combined FOM is the universe of data within a Combined Federation execution. A subset of this data is that which is to be shared among the security domains. A Combined FOM's classification level will be equal to or greater than the highest level of data to be processed by the Combined Federation[2].

2.2 Information Security Domains

Within a Combined Federation, secure information flow depends on each federation within the Combined Federation establishing an information security domain in which to operate and exchange data with other members in the Combined Federation. According to the *Department of Defense Information Systems Security Policy* [7] information security domains are:

- A set of *information objects* identifiable as belonging to a domain.

  The HLA prescribes that the set of information objects within a federation be documented in a FOM. Within the context of the HLA, the requirements of a federation can be applied to an information security domain. Information objects that will be shared (e.g., transferred) across a federation and among domains are documented in a Combined FOM. (A FOM containing the information objects common to the Combined Federation.) An information object whose ownership is transferred between domains must documented in the Combined FOM. A supplement to the OMT to document ownership transfer information is needed.

- A set of *members* (human and system processes) of a domain.

  The members of the information security domain are the individual simulation models, or federates that comprise the federation.

- An information domain *security policy* including:

  - Requirements for membership
  - Rules of access by members to information objects within the domain
  - Rules of import and export of information to/from the domain
  - Requirements for the protection of information objects.

  An information domain security policy is equivalent to a federation security policy. It should include the security requirements for membership in that information security domain. It could include physical and procedural security requirements, for example. The rules of access to information objects by member federates are embodied within the Object Model Template (OMT) and Run Time Infrastructure (RTI) Specification. The RTI enforces these rules of access within a federation according to the prescribed interface definition and associated protocol. The rules for import and export of information between security domains of different security levels are specified in a security policy document governing the Combined Federation . Other types of rules relating to information transfer (e.g., initialization files, after action review) also should

---

2 Data aggregation may cause the Combined FOM to be classified higher than the data contained in any single-level FOM.

be part of the Combined Federation security policy. Some additional protection mechanisms may be required for federation members (e.g., network encryption devices.)

2.2.1 Interdomain Security Information Flow Requirements

A federate in a high sensitivity information security domain may need to pass data to a federate within a lower sensitivity information domain, in which case we say there is a high-to-low information flow requirement. Conversely, information may be required at a high sensitivity level which has been generated at a low level, called a low-to-high information flow. Both of these information flows must be accommodated. In both cases, the flow must be in accordance with the information security domains' requirements and national information security policies. That is, only information that is properly of low sensitivity can be made available—directly or indirectly—to the low level federate.

The high-to-low flow must pass only information that is properly classified at the lower level. A high-to-low flow may pass a subset of the high information (that is less sensitive when divorced from its high context). For example, a simulation may involve a simulated missile hitting a simulated enemy tank; although the targeting technique may be classified, once the missile strikes the target, its existence and location become very visible and thus less sensitive. Alternatively, a restricted version of the high information may be more freely releasable. For example, data received directly from intelligence sensors may be classified, since they show the accuracy of intelligence collection; however, data expressed at a lower degree of precision may be less sensitive.

In all cases, high-to-low data flows must be screened (i.e., sanitized) so that only acceptable data is transferred. Rules for import and export of data to/from security domains must be developed during the federation design when it is known which objects are to flow between information security domains. These sanitization rules must be complete, unambiguous, and objective and they must be in a format for an automated security guard to enforce.

2.2.2 Intradomain Security Information Flow Requirements

A federation is assumed to function at a system high mode of operation, therefore, the information that is transferred within a federation may be accessible to all federates. No provision for the RTI to mediate access is planned or recommended at this time. Specific security requirements such as access control, identification and authentication, and audit are required security measures within a federate, but are not specifically discussed in this report. Encryption mechanisms may be required for information protection when federates are operating in a networked environment.

2.3 Assumption and Constraints

We have made the following assumptions concerning the composition of federation information security domains within the HLA. These assumption follow the *Department of Defense Information Systems Security Policy* [7] on information security domains.

- All information security domains have a fixed number of objects.

  A federation operates with a fixed set of objects that can be known to all members of the federation prior to federation execution.
- All information objects that are shared within a domain are equally accessible by all members of the domain.

  All information that is described in a FOM is releasable to all members of a federation.
- Each information object belongs to a single information domain.

  An information object transfer (object ownership transfer) is accomplished according to the federation security policies involved in the transfer. Additionally, the contents of an object residing in two separate domains may be the same or different.
- Interdomain object transfers are permitted if and only if there is one member (a security guard) common to both information domains who is permitted by policy of one domain (D1) to export (write) information to the other Domain (D2) and is permitted by policy of D2 to import (read) from D1.

  It is assumed that a security guard federate is the only path of communication between High and Low domains. The guard federate is physically and logically a member of two different federations, one in each domain. Acting on import and export (sanitization) rules specified in the Combined Federation security policy, the guard mediates all information flow between domains. The guard is essentially a multipolicy machine, supporting the security policy in all security domains [8,10]

2.4 Assurance

Because the functioning of the guard is crucial to enforcement of security in the mid-term architecture, accreditors want to be confident that the implementation is correct and complete. The guard must be strongly protected against tampering or malicious activity that would cause it to fail to enforce its security requirements.

Assurance for a guard is derived from its having a sound, conceptually simple, and unbypassable design, and its implementation corresponding to the design. The greater the risk (that is, the greater the difference between the high and low sensitivity levels) the more important assurance becomes. Assurance factors to be considered include the simplicity of the design and the development process and environment in which the guard was produced.

The HLA Guard will have to reside on a trusted B3 level [2] platform. The high level of assurance required for the HLA Guard is due to the perceived need to share top secret information down to unclassified within a Combined Federation. In the case of a top secret to unclassified Combined Federation, additional security requirements, such the user clearance level, will have to be considered as factors to mitigate security risk.

As more becomes known about the guard operational requirements within a Combined Federation, the assurance requirements and functionality trade-offs will be identified. Section 4 begins the process of identifying the HLA guard's operational requirements within the context of a Combined Federation and the RTI.

3.0 HLA SECURITY ARCHITECTURE

This report focuses on the mid-term architecture as it is considered to be the most viable for processing multilevel data within the next 3 to 5 years. The mid-term architecture, or the Secure Combined Federation Architecure is shown in Figure 2-1. The figure shows two federations operating in two physically separate information security domains. The architecture is extensible to more than two security domains. These domains are shown as Secret and Unclassified, although other security levels and compartmented information can be implemented. This concept of information security domains is consistent with the DoD Goal Security Architecture.[4] Federates within each information security domain are shown as 'lollipops.' A security guard resides between the two security domains to control the information flow in mutual accordance with their security policies. The HLA security guard federate is a member of each federation execution and performs some of the functions that a federate performs in order to operate as an active member of a *Combined Federation*. The term *security guard federate* is applied to the security and federate functionality discussed in this architecture.

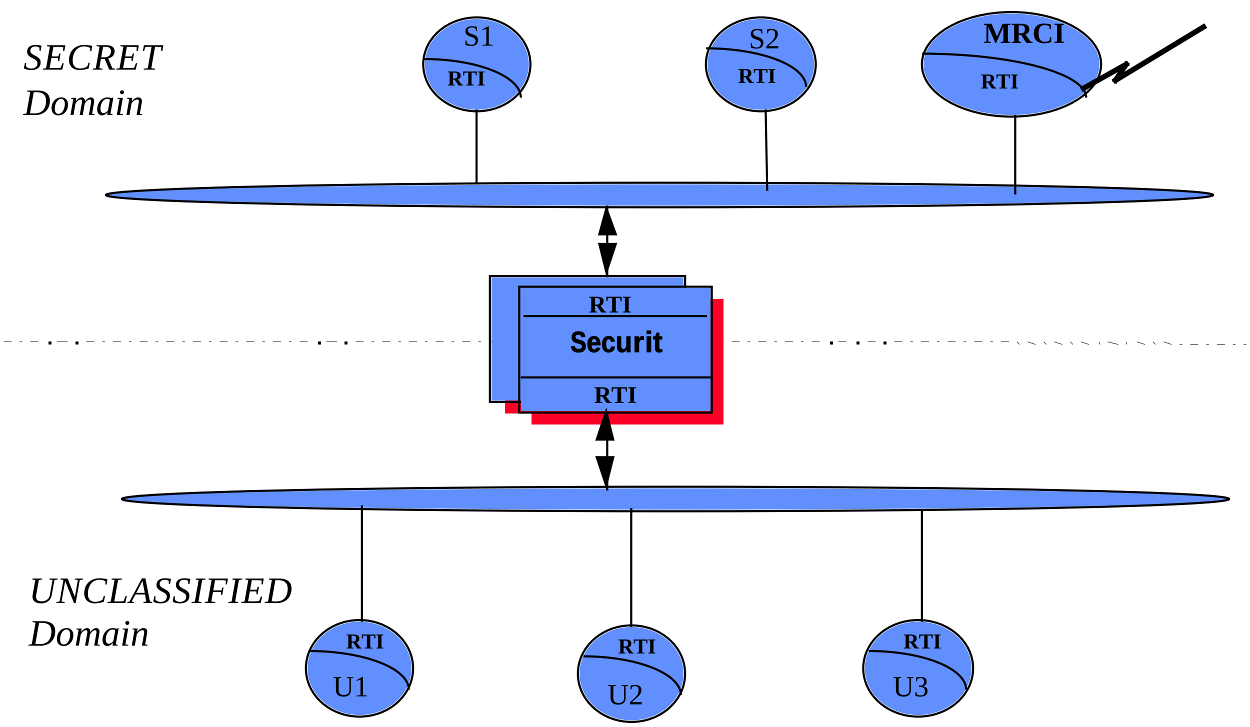

Figure 2-1. HLA Secure Combined Federation Architecture

The Run Time Infrastructure (RTI) is a component of the HLA whose function is to provide a set of standard services that allow federates to exchange data and exercise control information. Although, the internal design of an RTI is not specified as part of the HLA, the services and the interface it provides to the federates are defined in the HLA Interface Specification V1.0.[5] Figure

2-1 shows the RTI as distributed among the federates and as part of the guard. The fact that the RTI functionality may be distributed is not significant to this design.

The following assumptions have been made concerning the characteristics of the Secure Combined Federation architecture.

- Security within a federate and among its private or local federate data sources must be addressed by the individual federate (e.g., access controls, audit)
- External connections (e.g., the Modeling and Simulation Resource Repository [MSRR]) are not part of the runtime simuation exercise. The MRCI, as shown in the figure is viewed and interacts with the federation as a federate.
- Integrity of data in transit will not be undermined by the RTI and the communication mechanisms.
- Classified information that is transmitted outside an information security domain must be encrypted.
- The RTI processes all data exchanged among and used by more than one federate.
- All communications crossing security domains will be directed through the guard.
- All communication crossing security domains is documented in a Combined FOM.
- There is one RTI within each security domain.
- Each federate has a security policy and each federation has a security policy. (The contents of the federation security policy was described in Section 2.1)
- There is a global security policy for the Combined Federation containing the information that is to be imported and exported from each federation.

4.0 HLA SECURITY GUARD FUNCTIONALITY

This section contains a functional overview of the HLA security guard. The guard functional description in the following paragraphs provides a high level view of the types of operations it must perform. Next, the interactions of the guard with the RTI are described. The RTI Interface Specification is analyzed, and the action required by the guard in each interface call is noted. As a result of this analysis, some security issues have surfaced and are described at the end of this section.

4.1 Security Guard Functional Description

The HLA security guard federate is an automated process with the capability to downgrade or sanitize fixed, formatted data transferred from a higher security domain to a lower one. Information also flows from a lower security domain to a higher one.

The HLA automated security guard must perform several functions to support the bi-directional traffic flow. The *Message Guard Assessment* [11] contains the functional requirements for automated message-based guards. While the HLA security guard is not message-based, some of the functional requirements for those types of guards are applicable to the HLA security guard.

- User Interface Functions. The guard must provide a user interface to review erred data, resolve data errors, change add or delete sanitization rules, and administer the system. The guard may need a user interface feature that allows it to operate in manual mode to transmit image data, for example. (Due to inherent security vulnerabilities with image data, it cannot be automatically downgraded. [13])

- Error Processing Functions. The guard must ensure that data are not inadvertently mishandled within the system or routed to another system that is not cleared to process, store, or transmit the data. The guard requires the capability to detect and correct data that are in error.

- Security Functions. The guard contains security features. The guard will rely on the trusted operating system to support these features. These functions include: audit, accountability, and access control. The audit system must have the ability to record system, operator, and data processing events. A trusted path within the operating system will support accountability through identification and authentication functions. Access control mechanisms will permit users such as operators and administrators to access only relevant data to perform a particular function. Specific security authorizations (e.g., two-person control) may have to be supported by the operating system for changing a sanitization rule.

- Communication Functions. The guard must have the ability to manage all input and output communication channels. Minimally, two communication channels (one for input and one for output) must be supported.

In HLA context, the security guard is a member of all security domains in a federation execution and operates in near realtime. It has three major functions to support HLA federation executions:

- Map and process data and events from the RTI in one domain to the RTI in other domains. This process entails mapping of object identifiers in one domain to object identifiers in other domains. For example, each single level federation has its own "execution name." The guard/federate must track all the execution names and data in the Combined Federation belonging to that federation security domain. The guard federate also translates a service request from one RTI to the appropriate service request in other domains. The software to perform this function must be developed specifically for the guard, but may be partially designed from federate 'middleware' needed for any federate to interface the RTI.

- Sanitization of information in higher sensitivity domains to lower sensitivity domains. This process requires a unique rule set created specifically for each Combined Federation Execution. We assume this rule set can be described using a precise format that will enable the combined federation designers to configure the guard with a set of parameters. Other security guards have been developed with this feature. [12,15]

- Checking data type and format from lower sensitivity domains to higher sensitive domains is another unique security function performed by the guard. Information flowing from lower sensitivity domains to higher domains must be verified that it conforms to a known and expected format. To some extent the security guard must protect the higher level domains from data corruption by lower level domains.

Figure 4-1 shows a view of the three major functions performed by the HLA security guard federate.

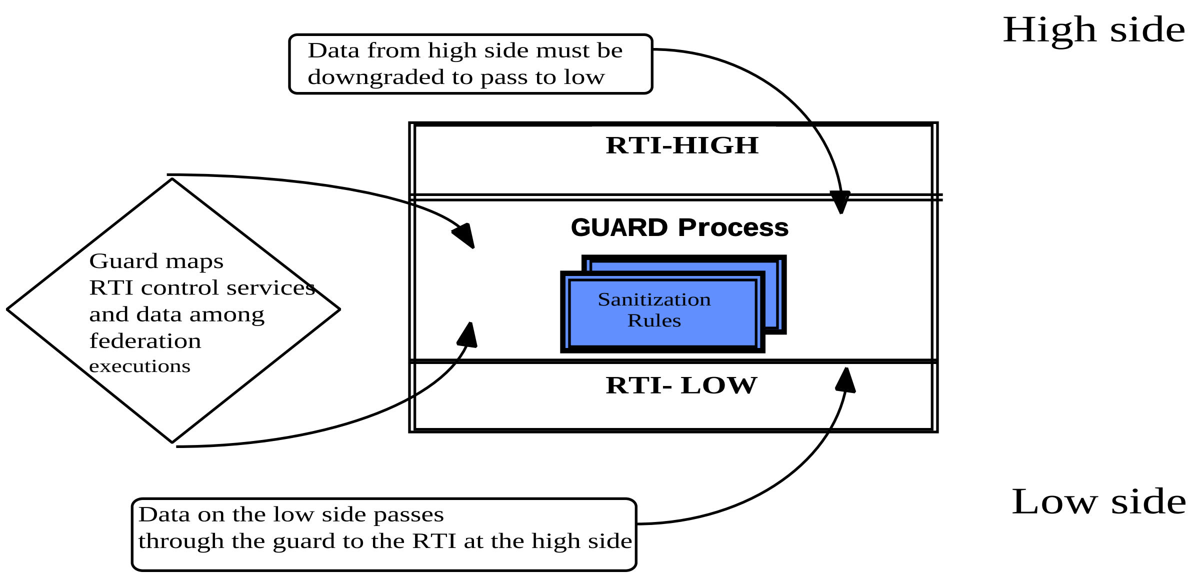

Figure 4-1. HLA Security Guard Federate Process

The analysis of the RTI interactions with the guard, in the following paragraphs, touches on all three of the security guard functions designated in Figure 4-1.

4.1.1 Security Interpretation of RTI Interface Services

The HLA Interface Specification [6] is organized into six logical groups: Federation Management, Declaration Management, Object Management, Ownership Management, Time Management, and Data Distribution Management. The following sections list each of the RTI service requests, describe their functionality, and indicate the security issues and guard interactions associated with the command. A table of service requests and guard interactions is given for each RTI service grouping.

4.1.1.1 Federation Management

Federation management services provide control over a federation execution. These services may be issued by the RTI in some cases, or a federate manager in other cases. The federation will designate a federate manager. The federation management services listed below are annotated with security implementation and operation notes.

Table 4-1 Federation Management

RTI Service	RTI Service Description and Security Notes
Create Federation Execution	Provides parameters to the RTI so it can build a new federation execution.
	1) A new parameter for security classification should be added to this service.
	2) The guard must join each of the single level federation.
	3) Each single level federation has its own "execution name". The guard/federate must maintain a mapping of all the execution names in the Combined Federation.
	4) The guard needs to maintain a mapping of all data from the single-level federates that will require transfer among security domain within the Combined Federation execution.
Destroy Federation Execution	Remove the named federation execution. All federation activity should have stopped and all federates should have resigned.
	1) Each single level federation must resign from the Combined Federation.
	2) The guard must resign from each federation execution.
Join Federation Execution	Affiliates the federate with the federation execution; declares important properties about the federate that determine classes of service and constraints.
	1) A federate can only join a federation at its same security domain.
	2) A federate may join more than one federation execution provided they are all at its same security level.
	3) The guard federate joins each federation participating in the Combined Federation. Only the guard federate is a member of federations at more than one security level.
Resign Federation Execution	Cessation of federation participation; ownership of attributes must have been resolved.
	1) The guard needs to know when a federate ceases to be part of the federation execution and security sensitive objects receive new owners, or if they are deleted.
	2) When a federate resigns from the federation execution, it must find owners for all its objects, or have the privilege to delete them. The new object attribute owner may be a federate in another security domain.
	3) A guard federate would resign from a single level federation when all the other federates have resigned and there are no federates at other security levels which are subscribed to an object published by a federate at that security level. The guard federate needs a signal from the RTI(s) that lets it know other federates within a domain have resigned.
Request Pause	Indicates a desire to stop the advance of the federation. The federate sends the RTI a pause-label and time that the RTI forwards to the other federates when it issues the Initiate Pause command.
	1) A single level federate would issue this request to the single level federation. The guard federate would Request Pause to each RTI in the Combined Federation on behalf of the initiating federate.

	2) A **request pause** within context of a Combined Federation could be interpreted as a combined federation-wide service (global), or it could pertain only to the federates in the single-level federation (local). There may be a need to distinguish local and global services.
Initiate Pause	The RTI instructs the federate to stop changing state at the specified time; otherwise the federate should stop as soon as possible.
	1) When a guard federate receives this instruction from the RTI, it sends a Request Pause to all RTIs in the Combined Federation.
Pause Achieved	Indicates that the federate has successfully paused at the federate time indicated.
	1) The guard federate sends this command to all RTIs in the combined federation.
Request Resume	Indicates the desire to resume the advance of the federation execution.
	1) The guard federate would issue this request to all other RTIs when it receives a **Initiate Resume** from the RTIs.
	2) A federate could manipulate these services to leak sensitive information from high to low.
Initiate Resume	The RTI informs a paused federate that it may return to the state it was in when it received the **Request Pause** request.
	1) See **Request Resume.**
Resume Achieved	Indicates that the federate is running.
	1) The guard federate would issue this service to all other RTIs when it receives **Initiate Resume** from the RTI.
Request Federation Save	Specifies that a federation save should take place. The RTI is sent the time, a save-label, and federate handle and federation execution name.
	1) A guard federate would issue this request to other RTIs when it receives an **Initiate Federate Save**.
	2) The guard federate should not have to save any state information about the Combined Federation.
Initiate Federate Save	The RTI tells the federate to save its state.
	1) See **Request Federation Save.**
Federate Save Begun	Tells the RTI that the federate is beginning to save its state.
	1) The guard federate requires an event to generate an event. This service is not initiated by an RTI event. 2) The guard federate might issue this service command to all RTIs immediately after it receives an **Initiate Federate Save** or when the actual time for the save has lapsed. 3) This could be an optional service for the guard federate to implement.
Federate Save Achieved	Tells the RTI that the federate has completed or failed its attempt to save state.
	1) See **Federate Save Begun**.
Request Restore	Directs the RTI to begin the federation restore. The RTI is sent the label supplied when a **Request Federation Save** service was invoked.
	1) A single level federate would issue this request to all RTIs in the combined federation.
Initiate Restore	The RTI instructs the federate to return to a previously saved state indicated by the save-label.

	1) The guard federate receives an **Initiate Restore** and issues a **Request Restore** to other RTIs.
Restore Achieved	Indicates that the federate has completed or failed to restore state.
	1) The guard federate needs to know when to issue this service command to all RTIs.

The federation management services do not call for the guard federate to perform sanitization of data. These services only require the guard federate to issue a corresponding RTI service when it receives a service request. There are two security concerns related to the federation management services. First, there is a signaling channel that can be exploited using the federation pause/resume and federation save/restore services. The channel could be exploited by a federate at the high level issues successive pause/resume services, each of which sends one bit of information to federates, or federation observers residing at the lower security level. The guard may have to detect these signaling conditions and audit these services. The second, which is less security relevant, concerns the guard acting as a 'gateway' between two or more federation executions. The guard federate should be designed with security as its primary function and 'gateway' functions secondary. As one of the next steps in the HLA guard federate design 'gateway' functions and security functions need to be delinated. The resolution to these and other security issues identified in the remaining sections will be resolved during the HLA guard federate design.

4.1.1.2 Declaration Management

The HLA requires a federate to declare to the RTI its capability to publish and subscribe object state information and interactions. The Combined FOM is consistent with these declarations. The guard federate acting a member of each federation execution will declare to each RTI in the Combined Federation its intent to generate and receive object state information and interactions on behalf of federates participating in each security domain within the exercise. Some objects, attributes, and interaction classes may be sensitive and cannot be shared across the security boundary. The guard appears to publish and subscribe to these object attribute values on behalf of federates in other security domains. The declaration management services in Table 4-2 are annotated with security design and operation notes.

Table 4-2 Declaration Management

RTI Service	RTI Service Description and Security Notes
Publish Object Class	Indicates the object attributes that a federate is capable of providing to the federation. Every object has an pre-defined attribute named privilegeToDeleteObject. (Note: More than one federate may publish values on the same object.)
	1) Publisher of the Object Class may not have privilege to delete. RTI enforces this access control over attributes within that class. Attributes within an object class may be published by different federates.

	2) The guard operations with respect to the delete privilege need to be fully specified. The guard will need to acquire the privilege to delete all objects that cross the security domain. It must both register and subscribe the privilegeToDeleteObject registered by other federates.
	3) The guard needs to know which federate in each security domain publishes which set of object attributes.
Publish Interaction Class	Informs the RTI which classes of interactions the federate will send to the federation.
	1) The guard needs to know which federate publishes which interaction classes.
	2) See **Publish Object Class**.
Subscribe Object Class Attribute	Declares by class which object attribute values the federate needs to discover and will receive for that class.
	1) The guard subscribes to only those object classes that must cross the security boundary. 2) A subscription to an object class implies subscription to all descendent object classes. The Combined FOM must be composed with attention to attribute values that dominate an attribute class.
Subscribe Interaction Class	Specifies the class of interactions which should or should not be sent to the federate.
	1) See **Subscribe Object Class Attribute**.
Control Updates	The RTI indicates to the federate that the specified attributes for the specified class are or are not required somewhere in the federation.
	The guard receiving this service request from the RTI must perform two actions. If the parameter supplied by this service indicates that updates are not required, the guard federate needs to inform the publishing federate to stop publication. If the parameter indicates that the specific object attributes are required then, the guard will have to issue a **Query Attribute Ownership** service to each RTI to find the publishing federate.
Control Interactions	The RTI indicates to the federate that the specified class of interactions is or is not required somewhere in the federation.
	1) See **Control Updates**.

The guard will be required to sanitize data to support the declaration management services. Since these services potentially support the transmission of data in both directions (low to high and high to low), the guard will be designed to ensure that unexpected data or data that is not formatted correctly will not be processed. The guard will enforce format consistency across the security boundary. This design requirement for the guard implies that single level federations must be implementing the same version of the RTI Application Program Interface (API.) In addition to the API syntax, HLA object semantics must be consistent. The guard will not be able to check for semantic consistency, so the process of making these objects consistent should be resolved as part of the single-level FOM and Combined FOM process.

Some object attributes will have more than one value in different security domains. Federates that operate at the higher levels may choose to subscribe to object attributes at the lower levels and maintain a classified and an unclassified version of the same object attribute. (Note: this requires that the federate to manage two data values for one attribute and this may require software changes.) Federates that operate at the higher levels will have to decide if two or more versions of ground truth are required. The federates operating at high will have the opportunity to know the unclassified, or lower level version of ground truth as well as a higher sensitivity of ground truth. The federations operating at the higher levels should make this determination during the Combined FOM development process.

4.1.1.3 Object Management

The RTI object management services support the creation, modification, and deletion of objects and interactions. Objects have identifiers which are held and distributed by the RTI. All object management services require an object ID as a parameter. Federates request the object ID and assign it to an object. Object IDs are not reusable.

Each single level federation will have a set of object IDs. The guard will have to maintain tables that map each object ID to its respective single level federation and object class. It is possible that different federations may use the same object ID for different classes of objects; or they may use different object IDs for the same class of objects. For each service involving an object ID the guard will perform a mapping from an object ID in one security domain to its representation in another domain. The following table contains each object management service and a short description of the guard's operation for each service.

Table 4-3 Object Management

RTI Service	RTI Service Description and Security Notes
Request ID	Request the federation execution-unique object ID numbers from the RTI. Each ID is valid for only one object registration.
	1) The guard federate will issue a request for object IDs for all objects that need to cross the security boundary. The request will be issued to each RTI in the Combined Federation Execution.
	2) The guard federate needs to maintain a mapping for objects that are the same across domains. For example, object ID x in one security domain may be known as object ID y in another domain.
Register Object	Links an object ID to an instance of an object class. All attributes specified as publishable by the instantiating federate are initially set to owned by that federate.
	1) The guard federate will register the objects are required to cross the security boundary to the appropriate RTI.
	2) The guard federate will appear to be the owner of all objects that cross the security boundary, even though it will not actually own any of the objects it publishes. (Note: the appearance of guard federate ownership applies to those federation executions reflecting the attribute, not the publishing federation.)

Update Attribute Values	The attribute owner uses this service to supply attribute changes to the federation through the RTI. Works in conjunction with **Reflect Attribute Values.**
	1) Data sanitization is required for information to flow across a security boundary from high to low. Bi-directional information flow require the guard to ensure the information is formatted correctly. Information flowing in either direction that is not formatted correctly is not transferred across the security boundary. Error conditions in the guard federate need to be specified.
Discover Object	Informs the federate that the RTI has discovered an object. An object is discovered by the RTI through a number of conditions.
	1) This service may lead to a transfer of data from one domain to others. Data may have to be sanitized.
Reflect Attribute Values	The RTI provides the federate with new reflected attribute values for an object. Works in conjunction with **Update Attribute Values**.
	1) Data sanitization is required for information to flow across a security boundary. See **Update Attribute Values**.
Send Interaction	A federate informs the RTI of an action taken by one object potentially towards another object.
	1) Affects the guard when an interaction occurs between federation executions. 2)The guard issues this command to the RTI which forwards it to other federates using the **Receive Interaction** request across the Combined Federation. 3) Data sanitization is required for information to flow across a security boundary.
Receive Interaction	The RTI provides the federates with information about an action taken by one federation object towards another object.
	1) Affects the guard when an interaction occurs between federation executions. 2) The guard receives this command from the RTI and uses the **Send Interaction** service to transfer data to other federates across the Combined Federation. 3) Data sanitization is required for information to flow across a security boundary.
Delete Object	The federate owning the object is deleting it from the federation execution.
	1) The guard must have the privilege to delete all objects that cross the security boundary.
	2) Only the object ID and not the actual data are passed across the security boundary. In these cases sanitization is not required.
	3) This service works with the **Remove Object** service.
Remove Object	The RTI is passing the delete object notice to other members of the federation execution. The object may be deleted because it is out of scope according to the data distribution rules.
	1) The guard receives this service from an RTI and sends the Delete Object service to the appropriate RTIs.
	2) Data values are not transferred.
Change Attribute Transportation Type	A federate chooses to change the transportation type associated with an object attribute.
	1) The guard passes this control information to the RTIs in other federations. 2) The RTI verifies that the federate owns the object attribute(s) that will change. The guard federate needs to appear to be the owner of all objects that cross from one domain to another, else RTI services that validate ownership will cause an exception.
Change Attribute Order Type	A federate chooses to change the attribute order type associated with an object attribute

	1) The guard passes this control information to the RTIs. 2) See **Change Attribute Transportation Type.**
Change Interaction Transportation Type	A federate chooses to change the transportation type associated with an interaction.
	1) The guard passes this control information to the RTIs. 2) See **Change Attribute Transportation Type.**
Change Interaction Order Type	A federate chooses to change the order type associated with an interaction.
	1) The guard passes this control information to the RTIs. 2) See **Change Attribute Transportation Type.**

Request Attribute Value Update	This service is used by a federate to stimulate the updates of values for specified attributes. The RTI will seek to obtain these attribute values from the appropriate federate.
	1) The guard issues this service request to other RTIs after it receives a **Provide Attribute Value Update.** The guard may have to issue this service to multiple RTIs since it may not know where the data resides in the Combined Federation Execution.
	2) See **Provide Attribute Value Update**.
Provide Attribute Value Update	The RTI initiates this service to request the current values for objects owned by the federate.
	1) The guard federate needs to know where the attribute update values are being generated. 2) The guard federate will request the data values from federates in other federation executions via the **Request Attribute Value Update** service.
Retract	A federate initiates an event retraction. (Used in discrete-event simulations to model interrupts.) Used by optimistic federates for anti-messages.
	1) The guard federate will forward a Retract to all RTIs once it receives a **Reflect Retraction** from an RTI.
	2) No data sanitization is required for this service.
Reflect Retraction	The RTI sends the retract event to those federates which need to 'cancel' some event.
	1) See **Retract**.

Some of the Object Management services will require the guard federate to perform sanitization. The guard federate, in order to perform its 'gateway function' may have to appear to be the owner of all objects that are transferred across a security domain. The guard federate will not actually own any objects, but may have to simulate object ownership.

4.1.1.4 Ownership Management

Ownership management services provide the ability for a federate to transfer ownership of object attributes. The owner is responsible for publishing and updating the object attribute value. The owner of the attribute does not imply the privilege to delete that attribute. The privilege to delete is a predefined attribute and is not transferred in the ownership management services.

Transferring object ownership from one federate to another federate across security domains can be achieved through the guard federate. The guard federate will appear to own the object attributes that are resident in other domains. The guard federate will not maintain a mapping of which federate owns a particular object attribute. Instead, to find out the real owner of an object attribute, the guard will query each RTI. When an object attribute requires transfer, the guard will locate that object owner from the RTI Management Object Model (MOM) [17] data. (We assume that object attributes requiring ownership transfer will pre-specified in the Combined FOM.) The Combined FOM should capture both the originating and destination domain. There are no data associated with ownership transfer, so sanitization rules do not need to be specified to implement an object transfer.

The following table contains each ownership management service and a short description of the guard federate's operation for each service.

Table 4-4 Ownership Management

RTI Service	RTI Service Description and Security Notes
Request Attribute Ownership Divestiture	Informs the RTI that the federate no longer wants to own the specified attributes of the specified objects. The federate can specify which federate should take ownership. The ownership transfer can be conditional, or negotiated.
	1) The guard federate will issue a request to the RTI for object attribute divestiture when it has received the **Attribute Ownership Acquisition Notification.**
	2) The Combined FOM should contain the information detailing the security domain that can be a candidate for receipt of ownership for each attribute.
Request Attribute Ownership Assumption	Informs a federate that the object attributes are available for ownership transfer. The federate returns the set of object attribute names that it is willing to own. This transfer must be confirmed by the **Attribute Ownership Acquisition Notification.**
	1) The guard federate will respond to this service by returning the object attribute name(s) that will be transferred to a federate in another security domain.
	2) The guard federate will error this service if the object ID is not valid.
Attribute Ownership Divestiture Notification	Notifies the federate that initiated the ownership transfer, that the transfer has been made. Lists the affected object attributes. A federate may receive multiple notices of ownership transfer. The federate should stop updating the attributes it no longer owns.
	1) The guard federate will receive this event after it has successfully transferred ownership of object attributes.
Attribute Ownership	The RTI confirms to a federate that it has ownership of the specified object attributes.

Acquisition Notification	
	1) The guard federate will receive this event after it has fully attained ownership of the specified attributes from **Request Attribute Ownership Assumption**.
Request Attribute Ownership Acquisition	Federate requests to RTI to own specified object attributes.
	1) The guard federate will request the attribute acquisition when it has received an event for an ownership transfer.
Request Attribute Ownership Release	Requests that a federate release ownership of the specified object attributes. Invoked as a result of **Request Attribute Ownership Acquisition**.
	1) The guard federate interprets this event as initiating an object attribute ownership transfer. 2) It locates the true owner of this object attribute through the **Query Attribute Ownership Service**.
Query Attribute Ownership	Query federate as to the ownership of an object attribute. The RTI returns the name of the federate that owns the object attribute, or if it available for ownership.
	1) The guard federate uses this service to locate an object attribute owner. This service is used when an ownership transfer has been initiated.

The ownership management services do not require data sanitization. We recommend that FOM developers do not classify object attribute names. If an object name is sensitive then the guard federate must perform checks and sanitize the object name and the object value. This results in additional performance loss at the guard federate.

The guard federate implementation of these services require that all ownership transfers across the security boundary be identified and documented in the Combined FOM.

4.1.1.5 Time Management

Time management services are used for controlling the advancement of federation time. Time advances must be coordinated with object management services to ensure that federates receive events that satisfy their exercise objectives. The time management services are intended to support federations with different timing requirements and may be used to support federates in the same federation with different timing requirements.

The time management services and the interoperation of these services with the guard have not been defined at this time. Some of the issues that must be addressed include:

- 1) Time synchronization between RTIs in separate security domains needs to be defined. The *RTI Interface Specification* may need additional services to accommodate separate federation executions.

- 2) Although the data transferred between domains using the time management functions are not security-relevant, the flow control implemented using these services can be subverted and used as covert signaling channels. The guard will have to audit these services to detect these channels.

- 3) The majority of the time management services originate from the federate. The guard, as a surrogate federate, needs an event to prompt a response and the time management services do not lend themselves to this type of execution scheme. For example, the guard federate does not have an event to prompt a **Request Federate Time** service.

The table below gives a brief description of the time management services. Security notes will be added in the future.

Table 4-5 Time Management

RTI Service	RTI Service Description
Request Federation Time	Requests the current estimate of the federate time. Federation time is the minimum of lower bound timestamp (LBTS) and the current value of the federates logical time.
Request LBTS	Federate requests current value of LBTS.
Request Federate Time	Federate requests value of federate's logical time.
Request Minimum Next Event Time	Requests minimum LBTS and the timestamp of the next time stamp ordered message that is held by the RTI for the next requesting federate.
Set Lookahead	Federate sets the current value of the federate's lookahead.
Request Lookahead	Federate queries the RTI for the current value of the lookahead for the federate.
Time Advance Request	Federate requests an advance of the federate's logical time.
Next Event Request	Federate requests the next Time Stamp Order message from the RTI.
Flush Queue Request	Federate request delivery of all messages stored in the RTI's internal queues and delivers them after invoking this service.
Time Advance Grant	RTI grants request for advancement of federate logical time.

4.1.1.6 Data Distribution Management

Data distribution management services provide a means for the RTI to distribute federation data efficiently. The concept of routing spaces is a multidimensional system that support a federate in either reading (subscription) or writing (update) data. The data distribution management services allow the federate to specify the routing space and then to associate an object attribute or interaction to that space. Routing spaces can be changed in the course of a federation execution.

If the data distribution management services are used in a Combined Federation, the guard, as a member of each federation execution will have to specify both the update and subscription regions to be equal to the allowable federation defined routing space. In terms of the guard, this has the effect of eliminating distribution based on the routing space. Figure 4-2 shows the relationship between the federates' regions and the guard's region. The guard's subscription/update region must encompass the entire routing space for the federation execution because the guard has no way of determining the boundary of a federate's region, or if a federate changed a region for a particular object attribute. In addition, the guard must be able to maintain a routing space for each federation execution in the Combined Federation.

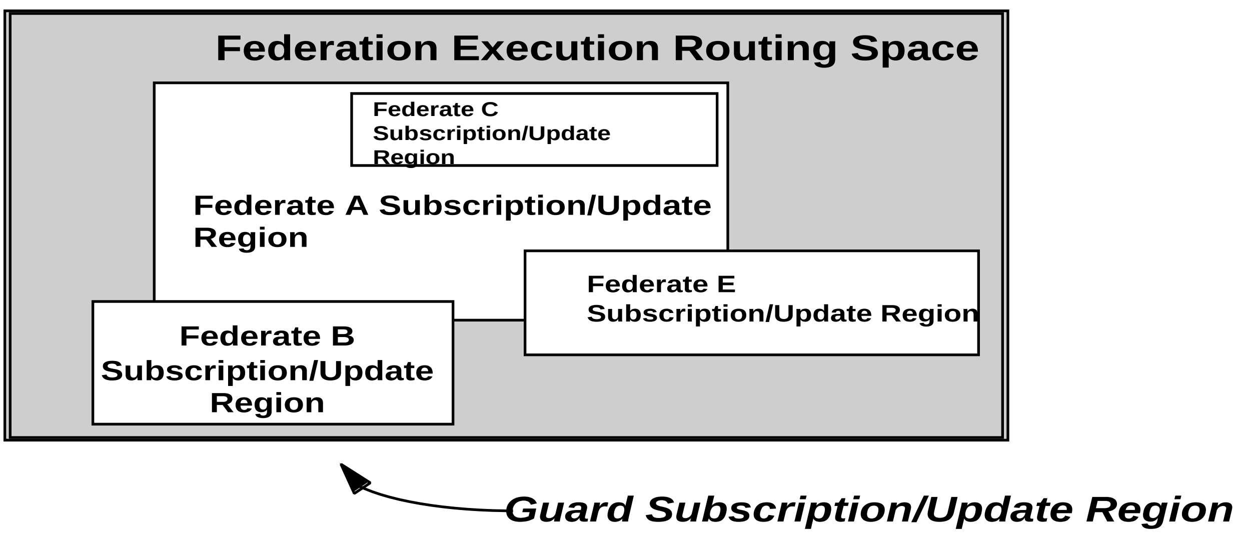

Figure 4-2. Guard Subscription/Update Region

Data distribution could not be effectively implemented through the guard with the RTI version 1.0. Additional RTI-to-federate services would have to be implemented if data distribution is required in Combined Federation.

The table below gives a brief description the data distribution management services. Security notes will be added in the future.

Table 4-6 Data Distribution Management

RTI Service	RTI Service Description
Create Update Region	Federate creates an update region within the routing space specified in the RID.
Create Subscription Region	Federate creates a subscription region. The RTI will only deliver data that falls within the bounds of that region.
Associate Update Region	The federate associates an update region with specific object attributes or an interaction class.
Change Thresholds	RTI informs the federate of new dimensions of the routing space. The federate may use this data to invoke **Modify Region** service.
Modify Region	Federate changes the bounds of the update or subscription region.
Delete Region	Federate deletes the update or subscription region.

4.2 Security Guard Design Issues

The interpretation of the RTI interface for the guard design identified several security design issues. These issues include specific RTI service implementation issues, architecture, the Combined FOM development process, and performance.

4.2.1 RTI Service Implementation

The HLA security guard acts as a gateway between two autonomous federation executions. The guard should be viewed as a transfer agent, which does not retain federation state information. If the guard were to implement time management services it would have to know which federates (in all federations) are using which time management scheme, and then be able to track time in each federate. To accomplish this task, the guard may have to duplicate the RTI logic and data storage. This is not a feasible solution, since the security guard's primary function is to securely pass data from one domain to another. To implement this level of time management control, we believe the RTI services may have to be modified to accommodate time management in a Combined Federation. Other solutions can also be considered. For example, federation time management schemes can be adhered to, but information originating from another domain can be executed serially as the RTI receives it. This solution may introduce data consistency issues, but trade-offs are needed in this area.

The HLA security guard is an event driven component. It needs a stimulus in order to respond. In some cases, there is no stimulus for the guard to respond to. This issue also raises the question whether some RTI services should be interpreted by the guard as Local -- to a single federation execution, or global -- to the Combined Federation. These and other RTI service issues identified in Tables 4-1 through 4-6 should be resolved as the architecture is further refined.

There should be a security requirement for identification and authentication (I&A) of federates within a federation execution. I&A is performed to confirm and verify that the federates in the federation are the entities they purport to be. The I&A process indirectly supports data integrity between a federate and the RTI. The RTI is the logical part of the HLA for performing this task. The RTI performs an identification exchange with the federates through the **Join Federation Execution** service. However, this service does not authenticate (e.g., require proof of its identity) the federate and therefore could be spoofed. The addition of a mechanism to authenticate a federate in the Join Federation Execution sequence will resolve this issue.

4.2.2 Architecture

Analysis of the RTI services revealed that for each incoming service request, the guard will have to output a service response to each RTI in the federation execution. The concept of the guard operating as an active federate in each security domain becomes more complex when the number of security domains is greater than two. In order to provide an extensible security architecture, more than two security domains should be part of the architecture. The design options for the guard federate include: (1) a guard configurable with multiple RTI interfaces, or (2) a guard to support only two RTI interfaces. The second option requires multiple guards to support federations with more than two security domains. There are trade-offs associated with both options.

Some of the security-related issues with more than two federations may affect the processing of data by a simulation model, because the data can be polyinstantiated at the higher levels; that is objects residing at the higher security levels will have multiple values differentiated only by security level. Figure 4-3 provides an example of federation views on polyinstantiated objects.

Polyinstantiated data occurs when the same object attribute is <u>instantiated</u> <u>and</u> <u>owned</u> by different federates at some point in time in different security domains. (Note: an object attribute can only be owned by one federate at a time.) For example, a federate in a Top Secret domain; with assistance from the guard, publishes values to federates at Secret and Unclassified domains. When the guard receives a **Reflect Attribute Values** request from a high security domain (RTI), it responds to this service with an **Update Attribute Values** to <u>each</u> security domain within the Combined Federation execution. At some point in the Combined Federation, the ownership of that object attribute changes to the Unclassified federate. The Unclassified federate publishes values at the Unclassified level to federates in both Secret and Top Security domains. If more than one federation has published this set of object attributes, then the federation at the highest security domain will have the most recent value from each domain. In this example, the Top Secret federate will have Top Secret and Unclassified values.

This issue does not impact the guard design[3], but it may impact the federate, or model design. Additionally, the federates receiving multiple views of the data will not have a security label associated with a view. This is another consideration for both the guard and federate design requirements.

Security Guard View of Combined Federation Data

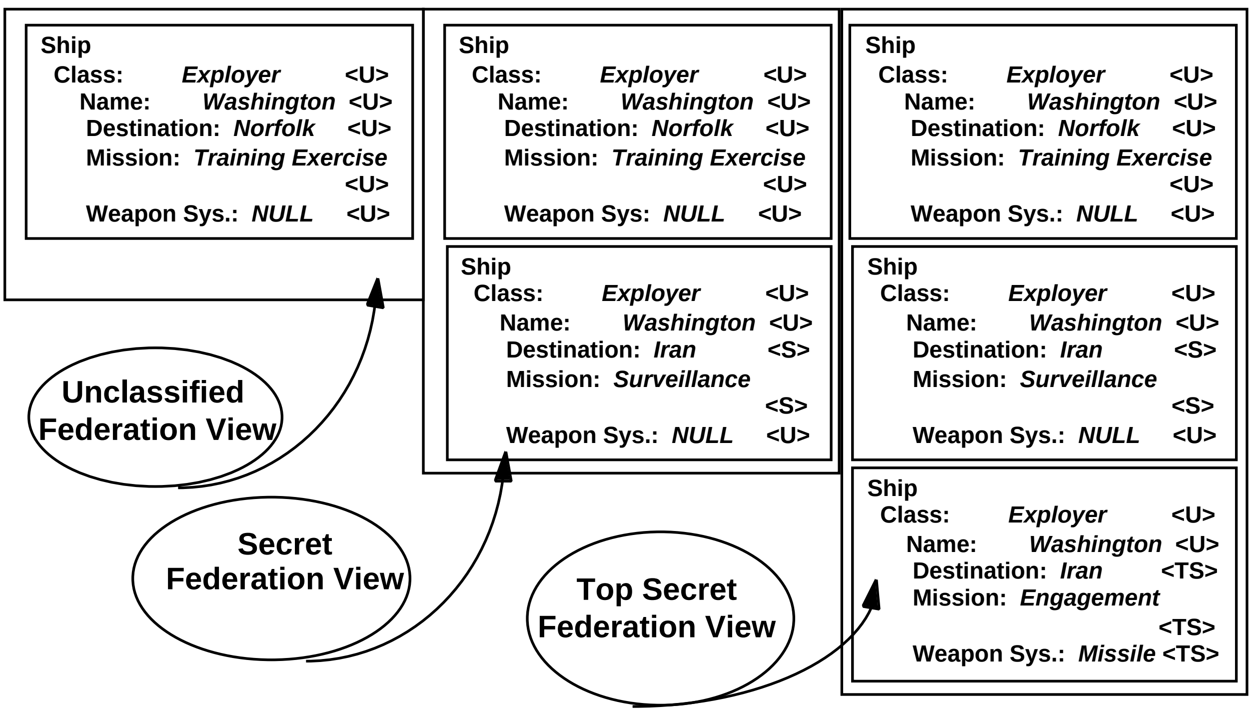

Figure 4-3. Polyinstantiated Data

Polyinstantiation of data more accurately reflects a multilevel view of the world. When a federate attempts to update attributes at a given security level, the value of that update must be maintained as well as the values for the object attribute at other security levels. All values need to be maintained in order to not compromise integrity. The high and low data values need to coexist and the guard and possibly the federates at higher levels will need the ability to track and manage the simultaneous occurrence of multi-valued objects. This is not a characteristic of federates as they currently operate. Note in Figure 4-3 that the Unclassified view and the Secret

---

3 The guard could be designed to implement 'cover stories' which would require it to identify a polyinstantiated object attribute, then convert that object attribute to different object. This would require more processing overhead and it would impact performance. This design option can be explored further if required.

view include the attribute for Weapon, even though it is null. This is required for data consistency across all security domains.

4.2.3 Combined FOM Development

The Combined FOM is the design document that represents the data set that will cross the security boundary between domains. The review of the RTI services raised some interesting issues in the development of a Combined FOM.

- RTI Services

  Object ownership transfer is one of the RTI services that will have to fully specified prior to federation execution. The guard will have to know: the from and to domains, object attributes that are transferred, and the from/to federates. Conditional transfers such as

  if attribute X=0, then Federate ABC may take ownership of X

  may be an operational requirement and may be possible for the guard to implement. This type of information must be specified in order for the sanitization rules to be applied and enforced. This must be documented in some format within the Combined FOM

- Labels

  Although the security architecture does not require labeled data, the sensitivity level of the data required in the Combined FOM may need a security label. We need to explore some options and determine the trusted mechanisms that can be used to address the labeling issue.

- Polyinstantiated Objects

  The Combined FOM developer should identify those objects where polyinstantiation is required. The Combined FOM developer has options as to permit polyinstantiation, but must recognize when and how it will occur. Federates must be able to process polyinstantiated objects.

- Combined FOM integration issues

  When two or more single-level FOMs are being analyzed or integrated for a Combined FOM naming conflicts, data structure type/scale differences and semantic differences are issues that must be resolved. In addition, all classes (data structure) for the Combined FOM must be completely defined.

- Security semantics

  Security semantics need to be defined for all Combined Federation executions. A generalized, simple set of security rules governing the interpretation of security in a Combined FOM will help the Combined Federation designer determine the object hierarchy. One such rule that can be made states: The security level of the object attributes can dominate the security level of the object class. This rule means that object attributes may have a higher security level then the object class. Another example relates to classification of object names. Object names should not be classified, only the object value. If object names were classified, then the guard would have to sanitize the object name as well as the object value. These and other rules should be developed according to the needs of M&S community who foresee the need to combine single-level federations. We are suggesting a set of security semantics in an OMT security extensions document.

4.2.4 Security Guard Performance

Performance is an important consideration for HLA guard design. However, it is difficult to predict performance in advance because of the following unknowns:

- Load on the guard: the number of calls and amount of data required to be passed between security domains
- Rules governing sanitization and downgrading
- Performance of the trusted platform
- Performance of the federate code
- Performance of the RTI middleware associated with the guard
- Performance of the underlying RTI and communications infrastructure, on which the guard will run.

A more in-depth analysis of the requirements for the Combined Federation architecture is needed to develop a set of engineering alternatives which can be prototyped to analyze the performance impacts of these issues on the guard. Continued work to evolve the security architecture based on user requirements and feedback is recommended. For example, a two (or more) guard configuration may become part of the architecture. HLA security and the architecture evolution is based on trade-offs between performance requirements, user security needs, costs, and the level of security risk of a proposed solution.

5.0 REFERENCES

1. U.S. Department of Defense, Defense Modeling and Simulation Office. *Primary Definition of the DoD High Level Architecture for Modeling and Simulation*, briefing slides, May 31, 1995. (Available online: <www.dmso.mil/dmso/wrkgrps/amg/amgbriefs>)

2. Department of Defense, *Department of defense Trusted Computer System Evaluation Criteria*, DoD 5200.28-STD. December 1985.

3. U.S. Department of Defense, Under Secretary of Defense for Acquisition and Technology,. *Modeling and Simulation (M&S) Master Plan*, October 1995. (Available online: <www.dmso.mil>)

4. U.S. Department of Defense, Defense Information Systems Agency (DISA), Center for Information System Security (CISS), Defense Information System Security Program, *DoD Goal Security Architecture* (Draft), V1.0, August 1993.

5. Filsinger, Jan, *System Security Approach for the High Level Architecture (HLA)*, 14th Workshop on Standards for Interoperability of Distributed Simulations, IST, Orlando Fl, March, 1996 (Available online: < www.dmso.mil/ >)

6. U.S. Department of Defense, Defense Simulation and Modeling Agency, *High Level Architecture For Simulations Interface Specification*, Version 1.0, September 1996. (Available online: <www.dmso.mil>)

7. Department of Defense Information Systems Security Policy, DSSP-SP.1, 22 February 1993.

8. Hilborn, Gene, *Information Domains Metapolicy*, 18th National Information Systems Security Conference (ISSC), October 1995, Baltimore, MD.

9. Fiorino, Thomas, et al, *Lessons Learned During the Life Cycle of an MLS Guard Deployed at Multiple Sites*, Eleventh Annual Computer Security Applications Conference, IEEE, December 1995, New Orleans.

10. Bell, Elliott, *Modeling the 'Multipolicy Machine*, Proceedings from the New Security Paradigms Workshop, August, 1994.

11. Defense Information Systems Agency (DISA), *Message Guard Assessment*, Technical Memorandum, DoD Multilevel Security Program, June 1994.

12 Operations Support Office (OSO), *Radiant Mercury Security Concept of Operations*, (FOUO) May 1995.

13. Cha, S. et al, *A Solution to the On-line Image Downgrading*, Eleventh Annual Computer Security Applications Conference, IEEE, December 1995, New Orleans.

14. U.S. Department of Defense, Defense Simulation and Modeling Agency, *Object Model Template Extensions* ,Version 0.3, May 1996. (Available online: <www.dmso.mil>)

15. Fiorino, T et al, *Lessons Learned During the Life Cycle of an MLS Guard Deployed at Multiple Sites*, Eleventh Annual Computer Security Applications Conference, IEEE, December 1995, New Orleans.

16. Dahmann, Dr. Judith, *HLA Federation Development and Execution Process*, 14th Workshop on Standards for Interoperability of Distributed Simulations, IST, Orlando Fl, March, 1996 (Available online: < www.dmso.mil/ >)

17. *HLA Management Object Model*, Version 0.2, 17 October 1996 (Available online: < www.dmso.mil/ >)